Multimedia Tools and Applications (2024) 83:34083–34108
https://doi.org/10.1007/s11042-023-16673-3

SAWIT: A small-sized animal wild image dataset with annotations

Thi Thu Thuy Nguyen[1] · Anne C. Eichholtzer[2] · Don A. Driscoll[2] · Nathan I. Semianiw[3] · Dean M. Corva[3] · Abbas Z. Kouzani[3] · Thanh Thi Nguyen[1] · Duc Thanh Nguyen[1]

Received: 16 April 2023 / Revised: 19 July 2023 / Accepted: 18 August 2023 /
Published online: 25 September 2023

Abstract
Computer vision has found many applications in automatic wildlife data analytics and biodiversity monitoring. Automating tasks like animal recognition or animal detection usually require machine learning models (e.g., deep neural networks) trained on annotated datasets. However, image datasets built for general purposes fail to capture realistic conditions of ecological studies, and existing datasets collected with camera-traps mainly focus on medium to large-sized animals. There is a lack of annotated small-sized animal datasets in the field. Small-sized animals (e.g., small mammals, frogs, lizards, arthropods) play an important role in ecosystems but are difficult to capture on camera-traps. They also present additional challenges: small animals can be more difficult to identify and blend more easily with their surroundings. To fill this gap, we introduce in this paper a new dataset dedicated to ecological studies of small-sized animals, and provide benchmark results of computer vision-based wildlife monitoring. The novelty of our work lies on SAWIT (small-sized animal wild image dataset), the first real-world dataset of small-sized animals, collected from camera traps and in realistic conditions. Our dataset consists of 34,434 images and is annotated by experts in the field with object-level annotations (bounding boxes) providing 34,820 annotated animals for seven animal categories. The dataset encompasses a wide range of challenging scenarios, such as occlusions, blurriness, and instances where animals blend into the dense vegetation. Based on the dataset, we benchmark two prevailing object detection algorithms: Faster RCNN and YOLO, and their variants. Experimental results show that all the variants of YOLO (version 5) perform similarly, ranging from 59.3% to 62.6% for the overall mean Average Precision (mAP) across all the animal categories. Faster RCNN with ResNet50 and HRNet backbone achieve 61.7% mAP and 58.5% mAP respectively. Through experiments, we indicate challenges and suggest research directions for computer vision-based wildlife monitoring. We provide both the dataset and the animal detection code at https://github.com/dtnguyen0304/sawit.

Thi Thu Thuy Nguyen and Anne C. Eichholtzer contributed equally as the first authors of this article.

✉ Duc Thanh Nguyen
duc.nguyen@deakin.edu.au

Extended author information available on the last page of the article

Keywords Real-world datasets · Small-sized animals · Animal detection · Camera traps

1 Introduction

Computer has been applied widely to many aspects of human life ranging from human-centred technologies to nature supporting services. With recent developments in imaging devices and techniques in computer vision, environmental studies are also benefiting from advanced algorithms applied to automatic wildlife data analytics or wildlife monitoring from camera traps [9]. Camera-trap systems record images or videos of animals and their behaviours using automatically-triggered cameras. They are a powerful tool for ecological studies, but often capture large datasets that can contain a high amount of false triggers. The analysis of wildlife camera data can therefore be time-consuming, but computer vision methods can help streamline the process.

There are two common tasks to analyse images of wildlife: animal recognition and animal detection. Animal recognition aims to identify animal species from an input image [13, 29, 30]. It implicitly assumes there is one animal per image, and that the animal occupies most of the camera's field of view [29]. In contrast, animal detection localises where the animals are in the image and identifies their types. Typically, results of object detection are represented by bounding boxes, each of which covers an animal and is associated with an animal type [47, 48]. Compared with animal recognition, animal detection is more challenging due to the presence of background and potential co-occurrence of multiple animal species in the same processed image. However, animal detection enables a wider range of biodiversity monitoring applications including animal counting, animal tracking and density estimation [30].

Recently, deep learning has been applied to advance both animal recognition and animal detection. To perform these tasks, deep neural networks are designed and trained on animal datasets accompanied with annotations. Annotations can be provided at image level (e.g., an animal class of an input image for the animal recognition task) or object level (e.g., bounding boxes of animals in an input image for the animal detection task). Object-level annotation is more labour-intensive than image-level annotation. To make deep neural networks robust and general under various conditions, training datasets are created to include variations of the data in practice (e.g., variations in animal appearance) and cover possible challenges (e.g., illumination changes, cluttered backgrounds). Datasets, therefore, are crucial to the success of deep learning techniques.

There is a large number of public datasets in the computer vision field such as ImageNet [6], PASCAL VOC [8], COCO [20] and Open Images Dataset [17]. However, those datasets serve general computer vision tasks, e.g., image recognition, object detection from natural images, and thus cannot always be used for ecology-relevant applications. On the other hand, datasets captured specifically by camera-traps (see Table 1) mostly focus on large- or medium-sized mammals and birds.

Small-sized animals (e.g., small reptiles, amphibians, mammals or arthropods) play an important role in ecosystems, but remain understudied compared to birds and mammals [3, 11, 38, 43]. Smaller fauna is generally more difficult to monitor. Despite some recent developments, most camera trap systems are still adapted to medium or large animals, and cannot reliably detect ectotherms or smaller mammals [22]. Small-sized animals also present additional challenges for automatic animal monitoring algorithms: they can be easily confused with the surrounding environment, especially in cluttered backgrounds such as dense vegetation and leaf litter; and can be more difficult to identify due to subtle differences in appearance.

In this paper, we provide a fully annotated dataset of small-sized animals to help fill the gap in wildlife image datasets. To this end, we make the following contributions in our work.

- SAWIT: a real-world dataset of small-sized animals. The dataset is collected in realistic and systematic conditions through camera traps equipped with motion detection based on pixel movement, across 30 woodland sites in southeastern Australia.
- Annotations for the collected data at object level on seven categories: frog, lizard, bird, small mammal, medium or big mammal, spider, and scorpion.
- A benchmark of prevailing object detection algorithms on the seven animal categories.

The remainder of the paper is organised as follows. Section 2 reviews a selection of existing animal datasets. Section 3 describes our proposed SAWIT. Experimental results of object detection algorithms on our dataset are presented and analysed in Section 4. Section 5 discusses future work and concludes our paper with remarks.

2 Related work

In this section, we review animal datasets published since 2012. Related animal datasets can be grouped into two categories: general animal datasets and camera trap-based animal datasets. General animal datasets are captured by popular cameras such as compact digital cameras, smartphones, and created for general purposes, e.g., pets collection, casual animal observations by non-specialists. Camera trap-based animal datasets, on the other hand, are captured by camera traps and created for a specific purpose, e.g., an ecological study focusing on specific type(s) of wildlife.

2.1 General animal datasets

Despite a large number of general animal datasets available, those with rich annotations remain scarce. Among them, a prominent annotated dataset of images of cats and dogs was introduced in [32]. The pet images were collected from four websites, including Catster, Dogster, Flickr and Google. Around 2,000-2,500 images were downloaded for each of 12 cat breeds and 25 dog breeds. These images were checked and filtered manually to create a dataset with 200 images for each of the 37 breeds. The dataset creates a benchmark for pet breed classification as it comes with rich annotations. Each pet is given a breed label as well as a pixel-level segmentation mask showing its body and a rectangle bounding box localising its head.

iNaturalist, introduced in [14] includes 859,000 images from over 5,000 different species of plants and animals. There were 9 animal classes, and the animals were annotated by bounding boxes. The dataset was used to evaluate Faster RCNN [35] in the task of animal detection.

Khan et al. [15] introduced an animal faces dataset, namely AnimalWeb, including 22.4K animal faces that cover 350 different animal species. Approximately 6K hours of experts and trained volunteers were spent for the design and development of the dataset. Around 200-250 images were downloaded from the image hosting website Flickr, and annotated by volunteers from the citizen science web portal Zooniverse. Annotation was performed on each face to identify 9 fiducial landmarks on key facial components such as eyes and mouth. The dataset is useful for various computer vision experiments such as face alignment, pose estimation and fine-grained image recognition [23, 33, 49, 50].

Recently, Ng et al. [27] introduced Animal Kingdom, a dataset that covers 850 species across 6 major animal classes in varied environmental conditions. The dataset consists of

Table 1 Public camera trap-based animal datasets, ordered by publication year. Note that small-sized animals covered in existing datasets are usually larger than our small-sized animals (e.g., spiders, scorpions)

Dataset	Year	Animal types	No. animal categories	No. locations	Annotation type	Size
Snapshot Serengeti	2015	Medium and large mammals: 25.67% Birds: 1.20% Small mammals: 0.01% Reptiles: 0.01% Empty: 73.11%	48	225	Image level	1.2 mil. images (322,653 with animals)
eMammal	2015	Mammals	15	NA	Image level	2.6 mil.
North American Camera Trap Images (NACTI)	2018	Medium and large mammals: 83.11% Birds: 2.11% Small mammals: 1.85% Empty/cars: 17.95%	28	5	Image level and object level (for 8,892 images)	3,367,383 images (training set)
Caltech Camera Traps (CCT)	2018	Medium and large mammals: 39.00% Birds: 4.08% Small mammals: 3.5% Reptiles: 0.12% Empty/cars: 53.22%	22	140	Image level and object level (for 57,868 images)	243,187 images (112,725 with animals)
Amur Tiger Re-identification in the Wild (ATRW)	2020	Medium and large mammals: 100%	1 (with 92 individual tigers)	10 (zoos)	Object level, keypoint, ID	8,076 videos
iWildCam Competition	2021	NA Empty: about 50%	206	414	Image level	263,528 images
Florida Wildlife Camera Trap	2021	Medium and large mammals: 65.73% Birds: 16.52% Small mammals: 0.5% Reptiles: 0.03% Negatives: 17.22%	22	2	Image level	104,495 images
SAWIT (Ours)	2023	Frogs: 24.67% Lizards: 11.37% Birds: 17.18% Small mammals: 4.62% Big mammals: 18.3% Spiders: 8.32% Scorpions: 15.55%	7	30	Object level	34,434 images

50 hours of long videos obtained from YouTube that can be used for video grounding. These videos were manually annotated on the framewise basis for both animal and action descriptions (with 140 fine-grained action classes given by expert annotators). Furthermore, the dataset comprises 33K frames with annotated animal poses.

2.2 Camera trap-based animal datasets

We summarise camera trap-based animal datasets published since 2012 in Table 1, and briefly describe them below.

Snapshot Serengeti [40] presents about 1.2 million image sequences collected over three years between 2010 and 2013. Volunteers from the general public provided labels for the collected images. Each image was labelled as with or without animal. A large portion of the images (73.11%) contains no animals. The remainder mostly contains medium and large mammals (e.g., wildebeests, zebras).

eMammal [24] is a 2-year project focusing on mammals. The project started in 2013 and collected 2.6 million images of medium and large mammals (e.g., bears, foxes, skunks, deer). Annotations were provided with different levels of details including object masks (foreground regions), appearance features such as body size, orientation, group size, and species types.

North America Camera Trap Images (NACTI) [41] is one of the largest animal datasets, covering 3.7 million images collected from five places in the USA. There are 28 animal classes in the dataset. A majority of the images (80%) contain large mammals (e.g., cattle, pigs, deer), and only 2% of the images contain birds. Images without animals represent 12% of the dataset. A mix of image- and object-level annotations is available (mostly for birds).

Caltech Camera Traps (CCT) was introduced in [1] and extended later with more data and animal types. This dataset contains 243,187 images collected from 140 different sites in the Southwestern United States. It mainly focuses on medium to large mammals like humans, cows, foxes, and dogs. Some images contain non-animal objects such as vehicles. CCT also provides a mix of image and object-level annotations, but only a subset of 57,864 images was annotated with bounding boxes.

Amur Tiger Re-identification in the Wild (ATRW) [19] contains over 8,000 video clips of 92 Amur tigers. This dataset includes high-resolution videos captured at multiple zoos in diverse lighting conditions and backgrounds. The tigers also appear in unconstrained poses. Time-synchronised surveillance cameras and tripod fixed single lens reflex cameras were used to collect the video clips. Annotation was performed on sampled frames and provided with bounding boxes enclosing the animals, key points on the animals' bodies, and identification of each individual tiger.

iWildCam 2021 Competition [2] was created to address the issue of population recognition of species collected from camera trap data. The training set includes 203,314 images captured from 323 locations, while the test set consists of 60,214 images collected from 91 locations. The dataset was composed of images from multiple sources, e.g., camera trap images, citizen scientists' images.

Florida wildlife camera trap [10] is a collection of 104,495 images with 22 animal categories. The dataset was collected from January 2018 to late 2019, in Corkscrew Swamp (Corkscrew) and Okaloacoochee Slough State Forest (OKSSF), in South Florida, the USA. It mainly contains records of medium and large mammals (66%) and birds (16.52%).

In general, the data in existing camera trap-based animal datasets are stored in two formats: still images and videos. Most of the datasets provide image-level annotations, e.g., [2, 10, 15, 24, 32, 36]. Those datasets are suitable for the animal recognition task. There are fewer

datasets offering object-level annotations, e.g., [1, 19, 21, 41]. Compared with image-level annotations, object-level annotations are more fine-grained. They thus require much more effort in data labelling, but are more informative and useful for a wider range of tasks, e.g., animal detection or animal counting. This review also highlights that existing animal datasets mainly focus on medium to large mammals and birds.

To enable research in automatic small-sized animal monitoring, we collect a new dataset of wild animals, with a focus on small fauna. Our dataset is captured by camera traps and in realistic conditions. Alongside the dataset, we provide object-level annotations for all collected animals. We note that the term "small" used in our paper refers to the size of animals in the real-world rather their size on captured images. Specifically, in a common convention of biology, small mammals are considered as species weighing less than 2 kg [18]. In this context, we call our focused animals (e.g., frogs, lizards, spiders, scorpions) "small" as they are extremely smaller than "large" animals, e.g., tigers, elephants, etc., studied in existing works (see Table 1).

3 SAWIT

3.1 Camera trap prototypes

Our camera traps were developed as part of a collaborative project among Deakin University[1], Arthur Rylah Institute[2], and citizen scientists affiliated with Land for Wildlife[3]. They are intended for day-and-night monitoring of herpetofauna (reptiles and amphibians) and small fauna [5]. Each camera-trap is designed to be mounted to a stake, about 30 cm above the ground. The lens is located at the bottom of the camera pointing towards the ground. The camera has a near-vertical orientation with a tilt angle of 15°; to exclude the mounting stake while still capturing a top view of small fauna. This ensures that small animals are within the focal range of the camera, and that their size can be consistently compared. The camera's field of view covers an area of approximately 24 (H) cm x 30 (W) cm (720 cm^2) on the ground.

To help direct animals towards the camera, two 5 m long plastic fences were installed on each side of the trap. The camera is powered by a lead-acid battery connected to a solar panel. The collected data is stored in a USB drive (up to 250GB for the study). Solar energy and large storage allow for long-term deployment of the system in the field. Figure 1 describes the prototype camera trap used to collect our dataset.

The camera is always on, and an on-board motion detection algorithm using background subtraction [45] continuously monitors up-coming frames for movement. The background subtraction algorithm estimates motion in the captured data, based on the number of changing pixels from a current frame to a background frame. If this number is higher than a predefined threshold, the current frame is considered to potentially present an animal and hence is stored. The background is also updated accordingly. In this study, camera traps were programmed to be highly sensitive to movement to maximise the detection of small fauna. A minimum threshold of $1cm^2$ of changing pixels per frame was chosen to reduce triggers by very small insects. The camera captures 25 coloured frames per second, with 640x480 resolution. An infrared light (940nm) illuminates the field of view in low light conditions. This enables the

[1] https://www.deakin.edu.au/life-environmental-sciences

[2] https://www.ari.vic.gov.au/

[3] https://www.wildlife.vic.gov.au/protecting-wildlife/land-for-wildlife

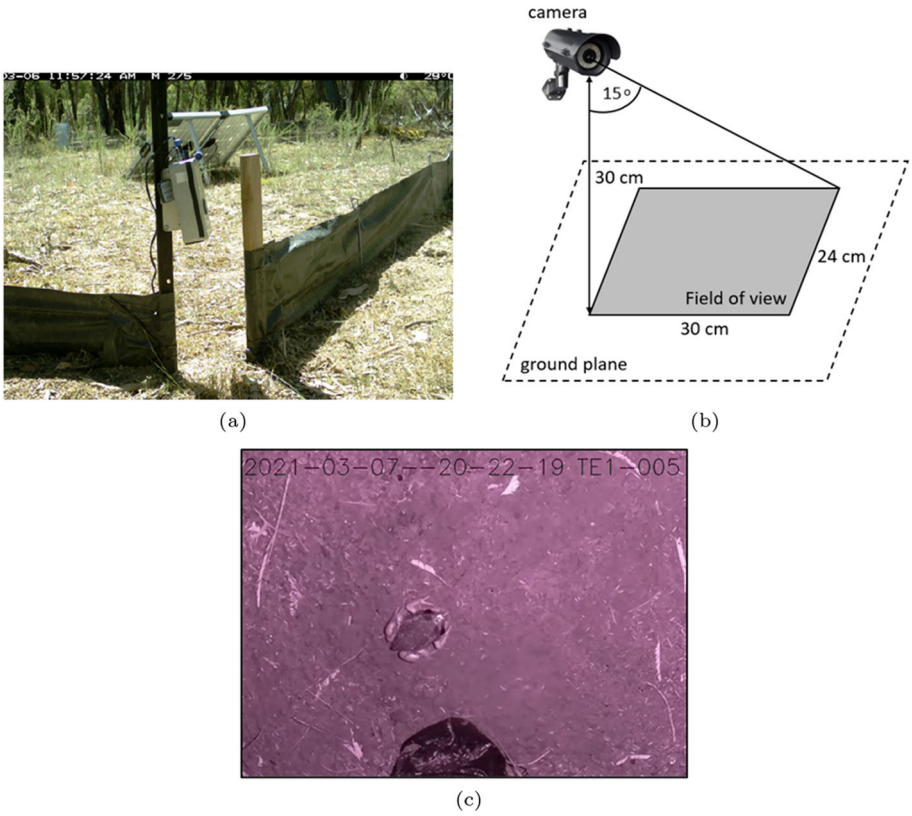

Fig. 1 (a) Prototype camera connected to a battery and a solar panel - pictured in the back, (b) camera set-up, and (c) a captured image

motion detection of endotherms and ectotherms during the day and night, in contrast with passive infra-red camera traps, which have limited capacity to detect ectotherms, particularly at night [7].

3.2 Field methods and data collection

Thirty (30) cameras were deployed on private lands around Ballarat, Daylesford, Castlemaine and Gisborne in Victoria, Australia. Each land included grassy dry or heathy dry forest vegetation [46]. The cameras were deployed for seven months between February 2021 and September 2021. Data were collected continuously (day and night), with some interruptions due to technical issues or lack of solar-power supply in the winter.

Images of animals collected in the wild using camera traps are a rich source of biological data yet also present challenges for computer-based algorithms [1, 10]. We illustrate the challenges of our dataset in Fig. 2. Specifically, animals captured by the camera traps are rarely perfectly framed, some animals may be too large for the camera and partially occluded (see Fig. 2(a)). The quality of the captured images is also strongly affected by environmental conditions, e.g., varying lighting conditions or weather changes altering the transparency of the camera lens (see Fig. 2(b)).

Fig. 2 (a) Some challenges of our dataset: (a) partial occlusion and close-up, (b) condensation in the top-left corner resulting in partial blurry, (c) a frog jumping through a frame resulting in a blurry capture, (d) animal blending in the background with soil and leaves, (e) a frog partially hidden in vegetation

Detection of small fauna brings its own set of challenges to the task of automatic animal monitoring. Small animals move quickly in open areas to avoid predation; the speed of these small animals can result in motion blur (see Fig. 2(c)). Small, cryptic animals can also blend into their surroundings, unlike most large mammals or birds that can easily be distinguished from the background. Within our area of study, lizards and frogs have green, brown, or gray colours. They can blend into the background of soil, rocks, and leaf litter (see Fig. 2(d)), or be covered by vegetation (see Fig. 2(e)).

Table 2 Edge cases in annotation of SAWIT

Edge case	Treatment	Example
An animal species that cannot be detected by a human annotator from a single frame but can be detected from a video sequence (i.e., using temporal information)	If the species can be detected by a human annotator from a sequence of frames, its images in all the frames are annotated.	 The frog highlighted in the picture is hard to be perceived visually from a single frame but can be detected from a short video sequence due its motion.
Annotation under occlusions	Animals whose width and height in the image (in 640x480-pixel resolution) are both less than 20 pixels are skipped.	 The highlighted box contains a part of a frog leg
Fast motion	Fast motion can be skipped if the animal involved in the motion is not clear to be detected.	 The moving region (highlighted) can be perceived but the animal species can not be identified.

3.3 Data annotation

We provide annotations for the collected data at object level. A random sample of 50,000 videos collected between February and August 2021 was manually filtered to select videos containing animals. We then manually extracted image frames containing animal(s) from each selected video. Finally, we annotated animal classes in the extracted frames at object level, by providing bounding boxes enclosing the animals. In this study, we focus on seven

Table 3 Detailed information of the collected animals in SAWIT dataset

Animal class	No. per-class images	% Images	No. per-class instances	% Instances
Frog	8268	24.01%	8591	24.67%
Lizard	3953	11.48%	3958	11.37%
Bird	5924	17.20%	5982	17.18%
Small mammal	1607	4.67%	1607	4.62%
Big mammal	6371	18.50%	6371	18.30%
Spider	2897	8.41%	2897	8.32%
Scorpion	5414	15.72%	5414	15.55%

animal classes including frog, lizard, bird, small mammal ($<$ 2kg), big mammal (medium-sized mammal), spider, and scorpion. The two arthropod groups were chosen because they have distinctive shapes and could demonstrate application of computer vision algorithms beyond vertebrates. We also observed animal types other than our interests (centipedes, beetles, ants, etc). Those animals were not annotated. The bounding boxes were made to fit the animals' bodies visible within the frame. There are several edge cases that required special consideration during the annotation. They are summarised in Table 2.

The annotation resulted in 34,434 images with 34,820 annotated animals. Frogs represent 24.67% of the overall number of animal instances. Small mammals are the rarest class, representing 4.62% of animal instances. We find there is usually one animal species detected per image. Note that, in this study, 'big mammals', include kangaroos and wallabies, but also refers to rabbits and echidnas which might be labelled as medium or even small animals in other datasets, e.g., Florida wildlife camera trap [10], snapshot serengeti [40]. We provide detailed information on the collected animals in Table 3 and show the distributions of bounding box sizes of the animals in Fig. 3. The distributions of the ratios of the bounding box sizes and image sizes vary across animal categories. Specifically, such distributions for frogs, lizards, spiders, and scorpions show single modes (less than 5%), while those for birds, small mammals, and big mammals include multiple modes (due to occlusions). In general, the animals' sizes in our dataset follow standard sizes in existing object detection datasets. For instance, the dominant sizes of objects in the PASCAL VOC [8] and COCO [20] datasets are less than 6% of the image's size. We illustrate several annotation results of our dataset in Fig. 4.

The output annotations are written in YOLO [34] and PASCAL VOC format [8] and stored in text files. Specifically, each frame containing animals is accompanied with an annotation file. Each animal detected in the frame is delineated by its closest bounding box, and written in the annotation file in either YOLO or PASCAL VOC format. For instance, let class_ID be the ID of the animal class of a detected animal. The class_ID varies in the range [0,6] and is defined as, 0 for Frog, 1 for Lizard, 2 for Bird, 3 for Small_mammal, 4 for Big_mammal, 5 for Spider, and 6 for Scorpion. Let x_{min}, y_{min}, x_{max}, y_{max}, width, and height respectively be the x- and y-coordinate of the top-left corner, x- and y-coordinate of the bottom-right corner, and width and height of the bounding box enclosing the animal. Depending on the format, the annotation information for the animal is written as follows.

- For YOLO format: class_ID x_{min} y_{min} width height.
- For PASCAL VOC format: class-name x_{min} y_{min} x_{max} y_{max}.

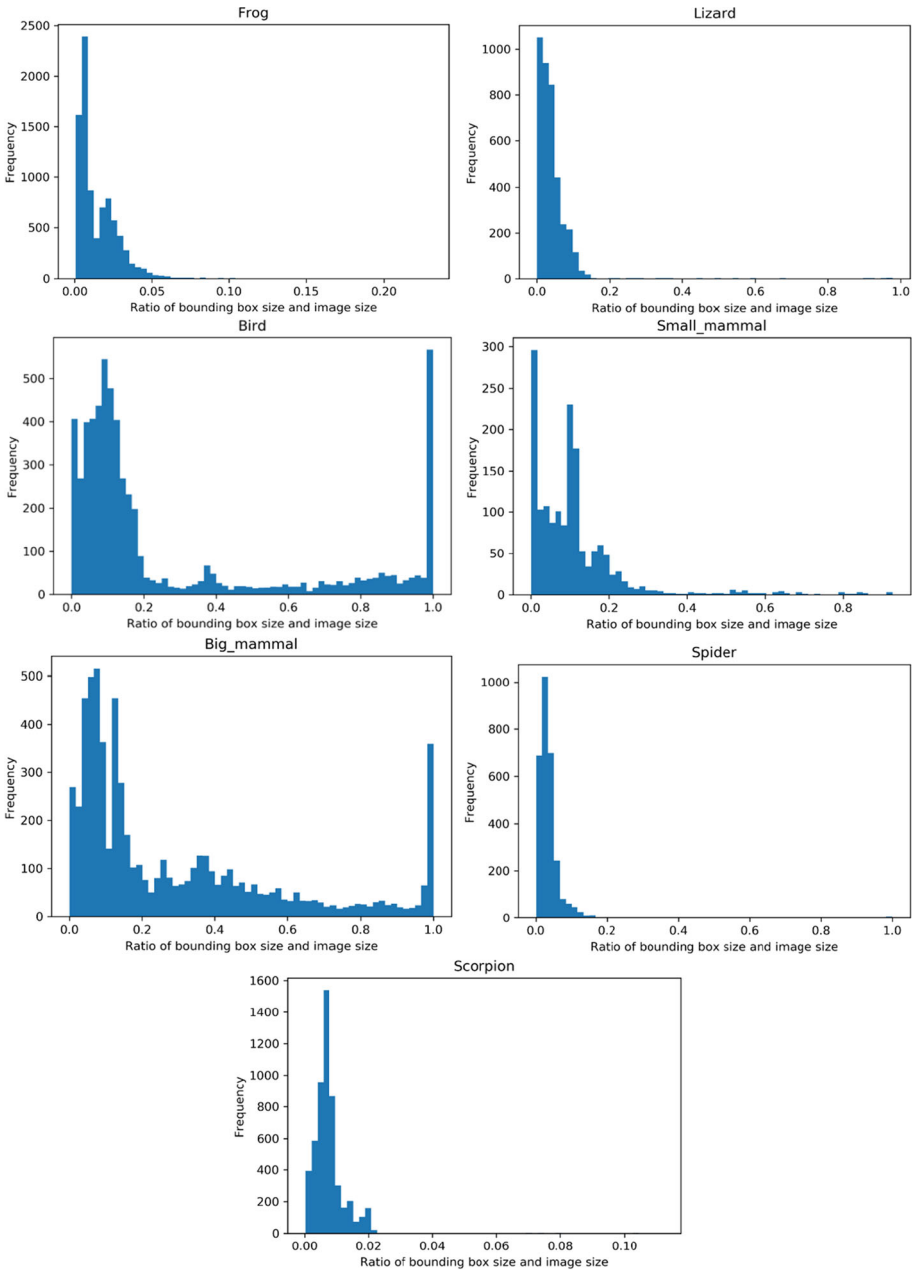

Fig. 3 Distributions of bounding box sizes in relation to input images' size) of collected animals in the SAWIT dataset. y-axis indicates the number of bounding boxes having a proportion of visibility falling within each range shown on the x-axis. Such variation is due to the variable sizes of the animals in the real-world as well as occlusions at image borders

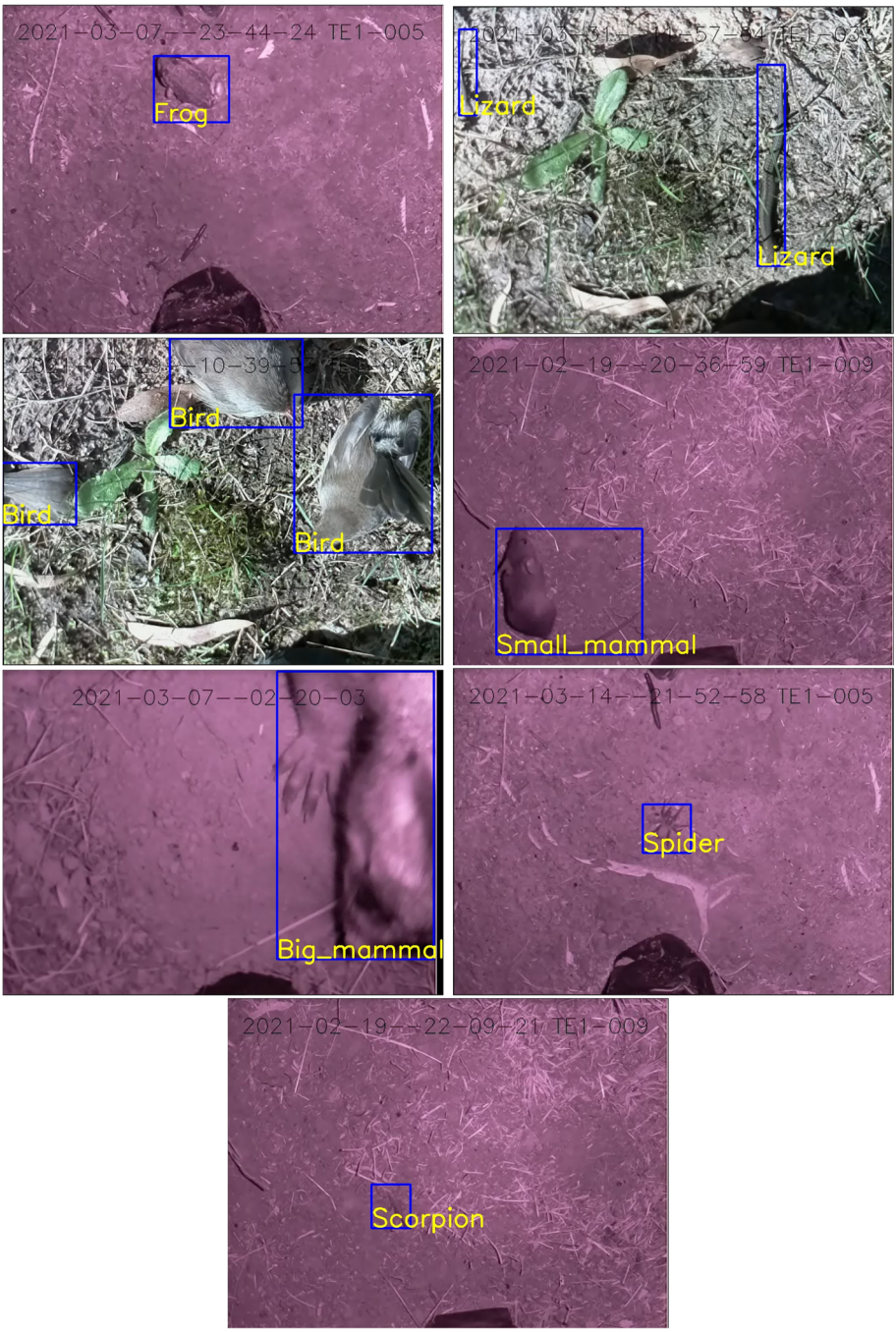

Fig. 4 Illustration of annotation results in the SAWIT dataset. The sizes and dimensions of bounding boxes (in blue colour) vary depending on animal types. Even within the same animal class, bounding box sizes also span across a wide range depending on the orientation of the animal in the image and the distance between the animal and the camera

Table 4 Details of YOLO's and Faster RCNN's architectures. We compare these architectures in terms of the Floating Point Operations Per Second (FLOPS), the number of parameters, and model size

Architecture	FLOPS (G)	No. parameters (M)	Model size (MB)
YOLOv5s	16.4	7	13.7
YOLOv5m	50.5	21	40.5
YOLOv5l	114.3	46.7	89.4
YOLOv5x	217.5	87.2	167
Faster RCNN (ResNet50)	3.8	25	314.9
Faster RCNN (HRNet)	4.3	9.3	207.5

4 Benchmark of object detection on SAWIT

4.1 Object detection algorithms

Object detection is a fundamental research problem in computer vision with a long history of development [51]. Object detection has been used for face detection [16], human detection [28], and plays a crucial role in a wide spectrum of applications. In this section, we provide a benchmark of state-of-the-art object detection on the seven animal categories collected in our dataset. Specifically, we experimented with two prevailing object detection algorithms including YOLO [34] and Faster RCNN [35]. Both methods are considered as state-of-the-art in object detection, and their capabilities have been verified on various detection tasks and datasets [51]. In biodiversity and conservation management, YOLO and Faster RCNN have been applied to detect koalas [4, 48], birds [47], fish [25, 26], and large mammals [37, 42]. In addition to high accuracy, these algorithms achieve real-time or near real-time performance. This is an advantage for wildlife data analytics tasks where continuous data processing is required.

Faster RCNN and YOLO follow two different mainstreams in object detection: proposal-based object detection (Faster RCNN) and proposal-free object detection (YOLO). The proposal-based approach divides object detection into two steps: proposal generation and object verification. The proposal generation step aims to produce bounding boxes which potentially contain objects of interest from an input image. These regions are called 'proposals'. In the object verification step, the presence of the animal is verified and the box is refined to fit with the objects (if any). The proposal-free approach simultaneously predicts an object's bounding box and its content (i.e., animal species), centred at every pixel or small region (e.g., a square area in YOLO) on an input image.

YOLO has a long developmental history with different versions. In this paper, we chose YOLO version 5, and adopted the publicly released Ultralytics repository [44]. YOLOv5 includes four architectures corresponding to four different scales: small architecture (YOLOv5s), medium architecture (YOLOv5m), large architecture (YOLOv5l), and extra-large architecture (YOLOv5x). These architectures have varying parameters (i.e., numbers of layers and filters) and hence results in different model sizes. We summarise YOLOv5's architectures in Table 4.

For Faster RCNN, we chose the mmdetection code library [31]. Faster RCNN can be adapted to different network backbones. In this paper, we investigated two commonly used backbones: ResNet50 [12] and HRNet [39]. Like YOLOv5's architectures, these backbones

Table 5 Train/test splits for all the animal classes in the SAWIT dataset

Animal class	No. training images (proportion)	No. test images (proportion)
Frog	4928 (59.60%)	3340 (40.40%)
Lizard	3045 (77.03%)	908 (22.97%)
Bird	3855 (65.07%)	2069 (34.93%)
Small mammal	1156 (71.94%)	451 (28.06%)
Big mammal	4476 (70.26%)	1895 (29.74%)
Spider	2108 (72.76%)	789 (27.24%)
Scorpion	3866 (71.41%)	1548 (28.59%)

vary in the number of parameters and model sizes. We present details of the variants of Faster RCNN in Table 4.

4.2 Results and discussions

YOLOv5 and Faster RCNN had to be adapted to detect the seven animal classes annotated in our dataset. We replaced the last layer in their architectures by a layer of seven nodes corresponding to the seven animal classes. The dataset was split into two subsets: training set and test set. We applied different ratios in the train/test splits to different animal classes, e.g., 59.60% of the frog images were used for training, but this number increased to 71.94% for the small mammal class. The different ratios used in the train/test splits were to prevent any similarities in the training and test data. Indeed, images in the training and test data are extracted from videos, and thus images from the same video share the same background. Videos collected at the same location also inherit the same surrounding environment. To avoid similarities in background information, the training and test images of every animal class were extracted from different videos, captured at different time steps and at different locations. Test images were also selected to cover the diversity of the collected data. Specifically, to reduce duplicates and similarities from continuous data capture, test images from the same video were sampled at discontinuous time steps. We present the train/test splits for all the animal classes in Table 5.

We trained all the variants of YOLOv5 and Faster RCNN using 50 epochs and 24 epochs, respectively. We set the learning rate to 0.01, batch size to 16, and made use of SGD optimiser for both YOLOv5 and Faster RCNN. Depending on the architectures, the training time varied from 3.6 hours to 25 hours. All experiments were implemented in Pytorch 1.10 and conducted

Table 6 Training and inference time of the experimented object detection models on the SAWIT dataset

Architecture	Training time (in hours)	Inference time (in No. frames per sec - fps)
YOLOv5s	3.684	250
YOLOv5m	5.413	142.8
YOLOv5l	7.516	83.3
YOLOv5x	13.153	47.6
Faster RCNN (ResNet50)	9.78	19.6
Faster RCNN (HRNet)	25	14.9

Table 7 Detection performance of YOLOv5 and Faster RCNN on the seven animal classes in the SAWIT dataset

	Yolo-v5s	Yolo-v5m	Yolo-v5l	Yolo-v5x	Faster RCNN (ResNet50)	Faster RCNN (HRNet)
All	59.3%	61.2%	62.6%	61.3%	61.7%	58.5%
Frog	64.2%	71.7%	61.6%	64.0%	76.8%	68.9%
Lizard	45.4%	47.7%	48.7%	46.8%	52.1%	39.0%
Bird	89.4%	90.2%	91.9%	90.5%	75.0%	69.2%
Small mammal	34.7%	34.9%	40.8%	41.8%	47.2%	52.6%
Big mammal	55.9%	59.6%	65.5%	55.0%	62.5%	65.9%
Spider	90.3%	90.6%	94.4%	94.7%	92.6%	92.1%
Scorpion	35.2%	33.6%	35.3%	36.1%	25.3%	21.7%

on 2 NVIDIA GeForce RTX2080Ti GPUs. We report the training and inference time of all the experimented architectures in Table 6.

We compared the object detection algorithms via the mean average precision (mAP) metric. True positives and false alarms were determined using the PASCAL VOC standard [8]. A detected object is considered as a true positive if there is a match in the ground-truth data. The match between a detected object and a ground-truth object is measured as the intersection over union (IOU) of the bounding boxes enclosing these objects. A match is confirmed if this IOU is greater than a threshold, which is set to 0.5 (a commonly used value) in our experiments.

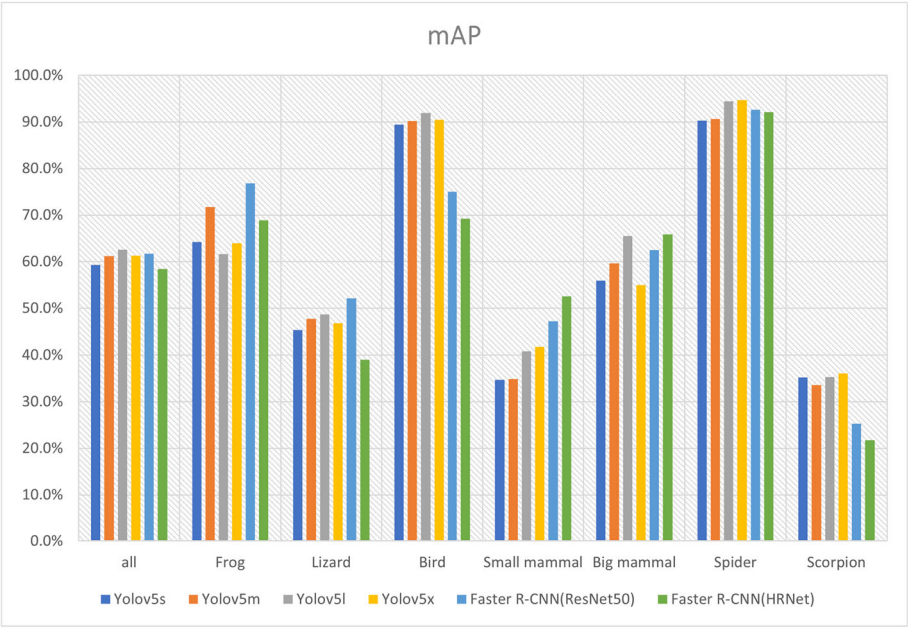

Fig. 5 mAP of YOLOv5 and Faster RCNN on the seven animal classes in the SAWIT dataset

We report the mAP of the object detection algorithms and their variants on our test set in Table 7, and plot the methods' mAP in a bar chart in Fig. 5. All the variants of YOLOv5 perform similarly (varying from 59.3% to 62.6% for the mAP on all the animal classes). The large model (YOLOv5l) achieves the best overall performance (with 62.6% mAP) across object detection methods and variants (see Fig. 5). Faster RCNN with ResNet50 ranks second for the overall performance. However, the difference between its mAP and YOLOv5l's mAP is marginal (about 1%). Faster RCNN with HRNet obtains the least overall performance (with 58.5% mAP), while requiring longer training time. This is probably because HRNet is designed to handle high-resolution images, and thus may not have advantages on our images captured in relatively low resolution (640x480). To have a more comprehensive evaluation, one should also consider both detection accuracy and computational speed. We present the processing time of all the detection algorithms and their variants in Table 6, and compare these algorithms regarding their detection accuracy (mAP) and inference time (fps) in Fig. 6. As shown in Tables 6, 7 and Fig. 6, YOLOv5l seems to optimise both accuracy and computational speed criteria.

Spiders can be well detected by all the detection methods (with more than 90% mAP, see Fig. 5). YOLOv5 stands out when detecting birds, achieving about 90% mAP. Scorpions appear to be the most challenging class for all the algorithms. Small mammals are also difficult to detect, probably due to limited training data. We found that all the variants of both the detection methods (i.e., YOLOv5 or Faster RCNN) perform consistently across the animal classes, with similar mAP ranking order of the animal classes for all the variants of each method.

To further investigate the object detectors, we report the loss and accuracy (in mAP) charts of the detectors (and their variants) in different training epochs in Figs. 7 and 8, respectively. YOLOv5 makes use of three different losses (Fig. 7(a)) including box regression, animal classification, and objectness detection losses (i.e., confirming whether a bounding box captures an object). In contrast, Faster RCNN utilises only box regression and animal classification losses (Fig. 7(b)). As shown in Fig. 7, all the detectors and their variants well converge. In addition, all variants of each detector perform consistently across loss functions

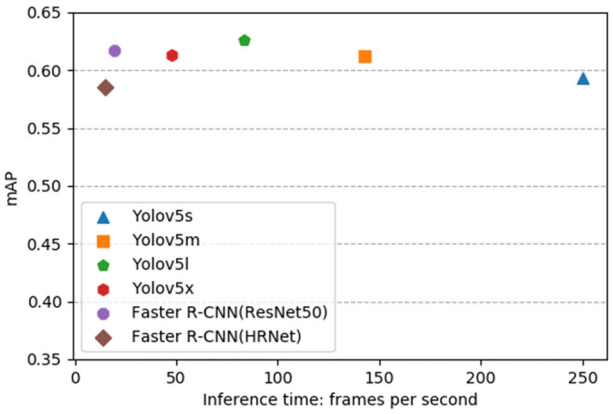

Fig. 6 Comparison of YOLOv5 and Faster RCNN in regard to detection accuracy (mAP) and computational speed in inference (fps)

(e.g., box loss, classification loss). The detection accuracy curves (Fig. 8), on the other hand, fluctuate during training but achieve best performances at 50 epochs (for YOLOv5) and 24 epochs (for Faster RCNN).

We illustrate successful detection results by YOLOv5 and Faster RCNN in Figs. 9(a) and (b), and 10(a) and (b). Some of the animals detected with high confidence by computer vision algorithms could be difficult for even humans to detect. For instance, Figs. 9(b) and 10(a) show cases where the animals and their surrounding background have similar colour and texture. Figure 10(b) presents a case where a lizard camouflages into a background of grass and leaf litter. We observed that Faster RCNN often gets extremely high confidence scores (e.g., 100%) in these cases, compared with YOLOv5.

We demonstrate challenges of our dataset (e.g., occlusions, cluttered backgrounds, shadows due to illumination conditions, fast motion) in Figs. 11, 12(a) and (b), 13, 14, 15, 16,

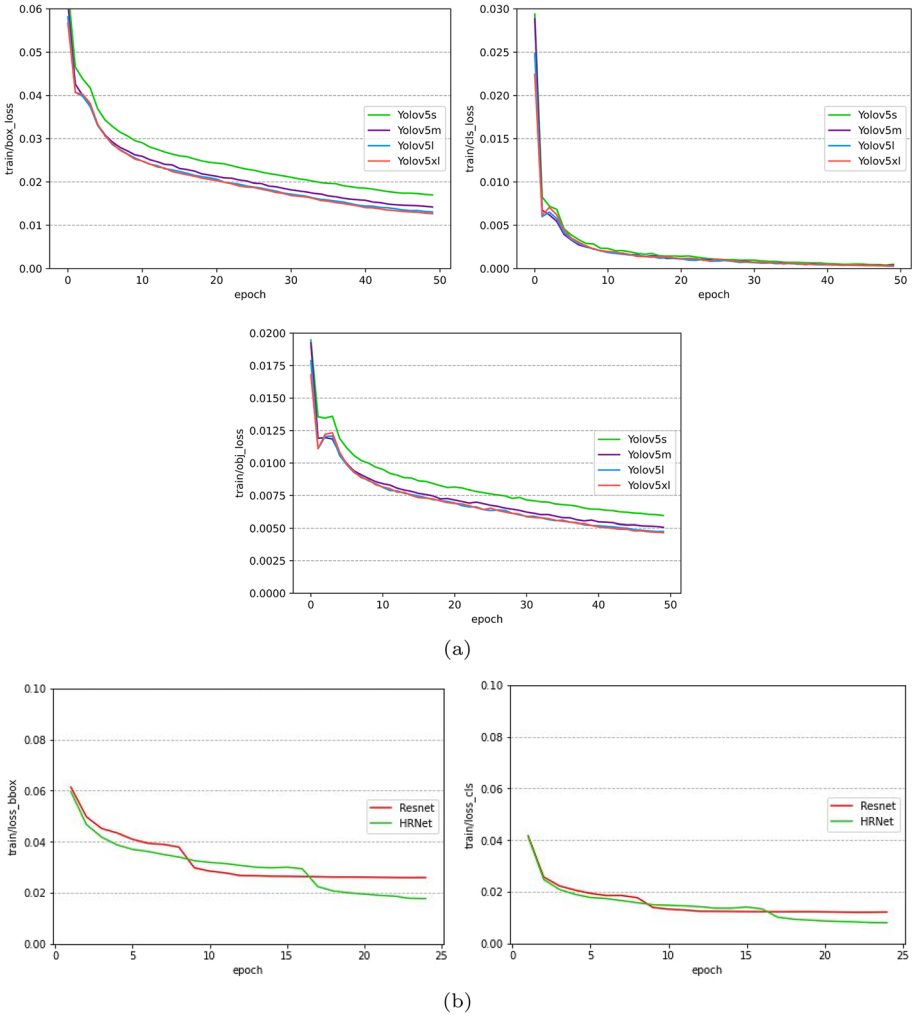

Fig. 7 Loss curves of YOLOv5 (a) and Faster RCNN (b)

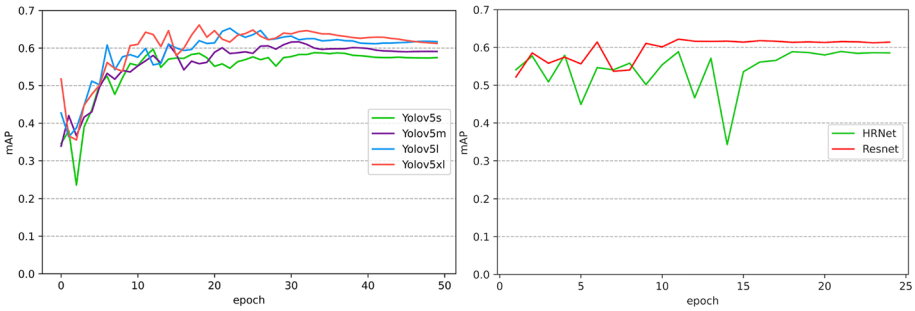

Fig. 8 Detection accuracy curves of YOLOv5 (a) and Faster RCNN (b)

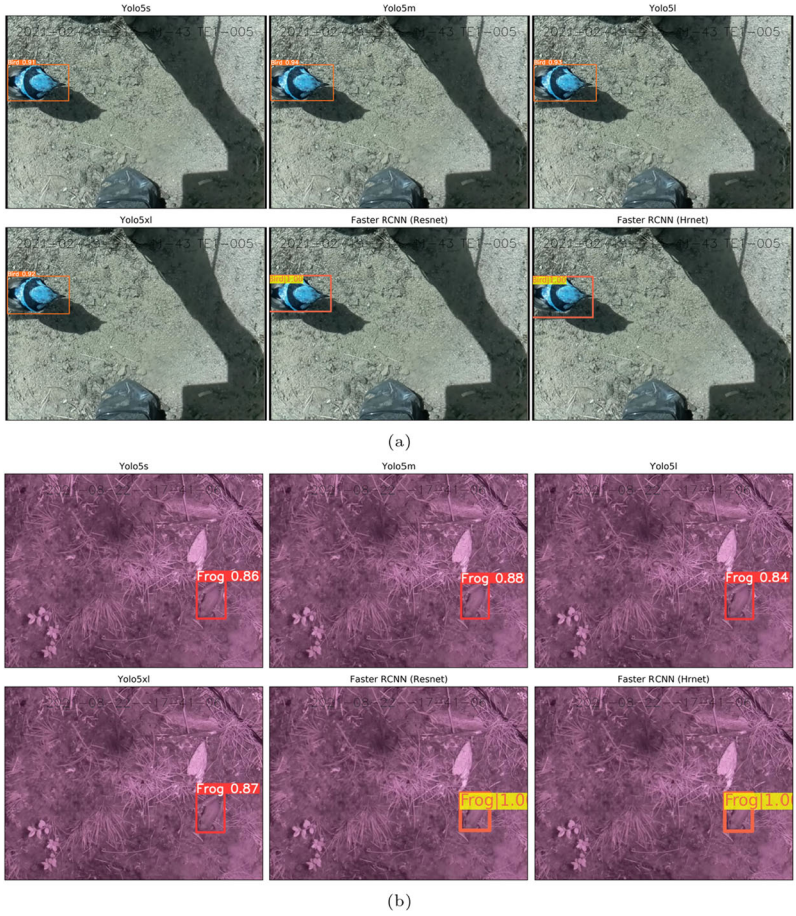

Fig. 9 Illustration of successful detections by YOLOv5 and Faster RCNN on the SAWIT dataset. (a) A clear image of a bird, which allows all detection algorithms to recognise the bird with high confidence. (b) Another successful example where a frog and the background have similar texture

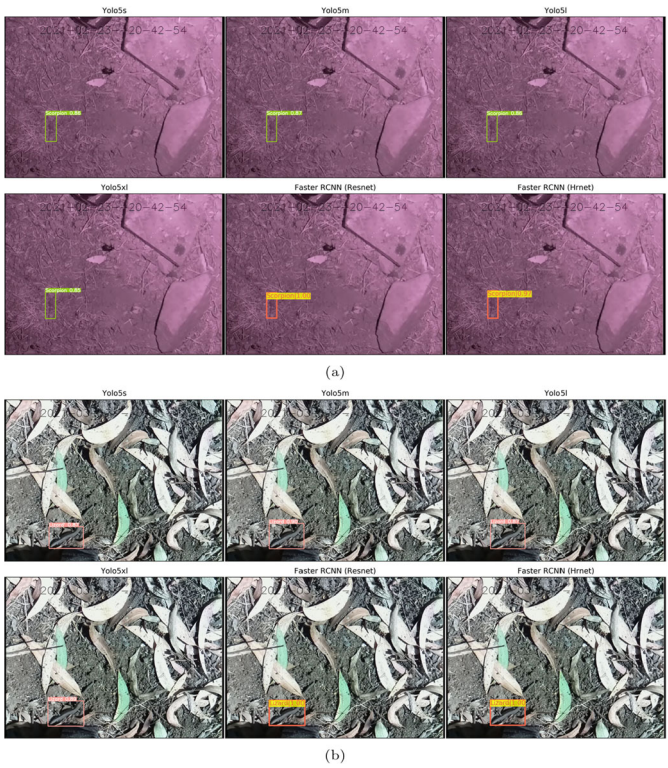

Fig. 10 Other successful cases of YOLOv5 and Faster RCNN on the SAWIT dataset. (a) A scorpion that looks like a twig. (b) A lizard blends into a background of grass and leaf litter

Fig. 11 A case where YOLOv5 and Faster RCNN models produce different detections. Variants of YOLOv5 yield similar results, which correctly detect the bird. In contrast, Faster RCNN models incorrectly identify the object as a big mammal

through which we discuss the performance of both YOLOv5 and Faster RCNN. In some cases, YOLOv5 and Faster RCNN produce discrepant class labels for the same animal. For instance, the bird in Fig. 11 is recognised differently by YOLOv5 and Faster RCNN. We found that Faster RCNN is more sensitive to occlusions, compared with YOLOv5. In addition, Faster RCNN often generates multiple labels for the same object (see Fig. 12). Another challenging case to Faster RCNN is shown in Fig. 13, where the algorithm generates several false alarms in a cluttered background. Figure 14 illustrates a case where both YOLOv5 and Faster RCNN can detect a bird when there are shadows, but Faster RCNN makes better fitted results. Faster RCNN also seems to perform better than YOLOv5 in detecting animals in fast motion and in detecting tiny animals. Specifically, while both versions of Faster RCNN can

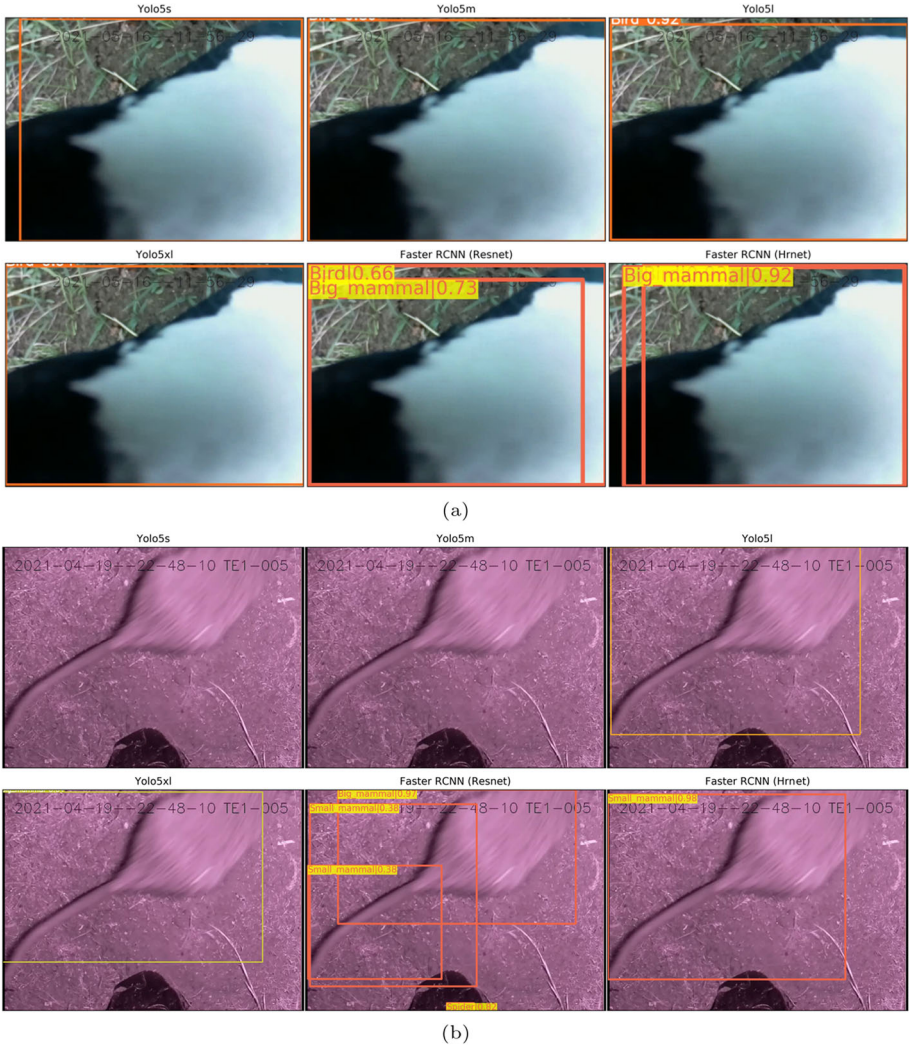

Fig. 12 Detection under occlusions. (a) A bird captured in close distance from the camera, resulting in severe occlusion. (b) A rat with the head occluded

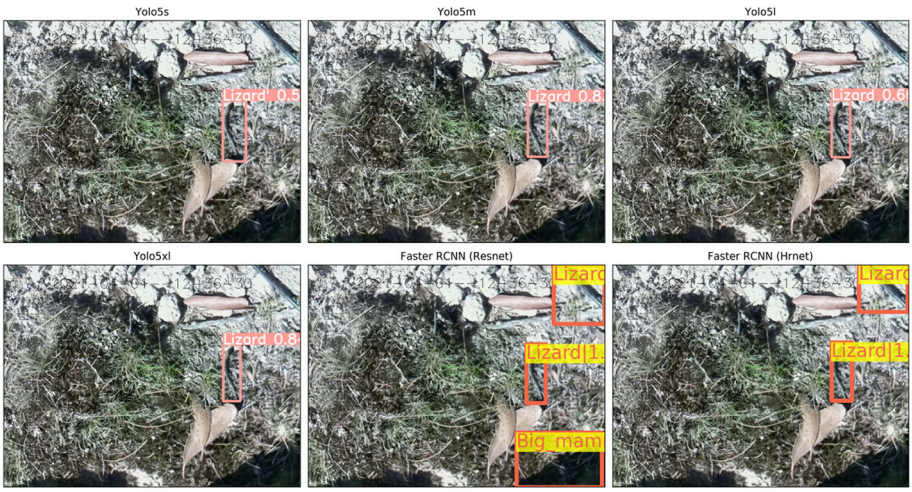

Fig. 13 Detection in a cluttered background. A lizard in a cluttered background with tree twigs. While YOLOv5 can well distinguish between the lizard and the twigs, Faster RCNN generates false alarms

detect a frog in fast motion (see Fig. 15), only YOLOv5m is able to do so. Faster RCNN with HRNet backbone appears to have an advantage in detecting very tiny objects as the HRNet is purposely designed to learn details from high-resolution images (see Fig. 16). These results show that SAWIT can be considered as a benchmark dataset of small-sized animals, covering various practical challenges for camera trap-based animal monitoring research.

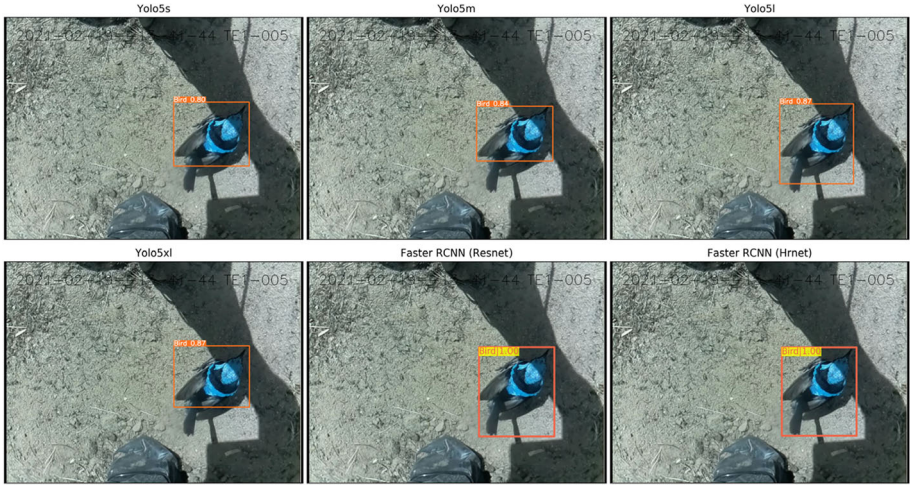

Fig. 14 Detection under shadows. Although shadows do not hinder the detection capability of both YOLOv5 and Faster RCNN, detections produced by Faster RCNN better fit the animal

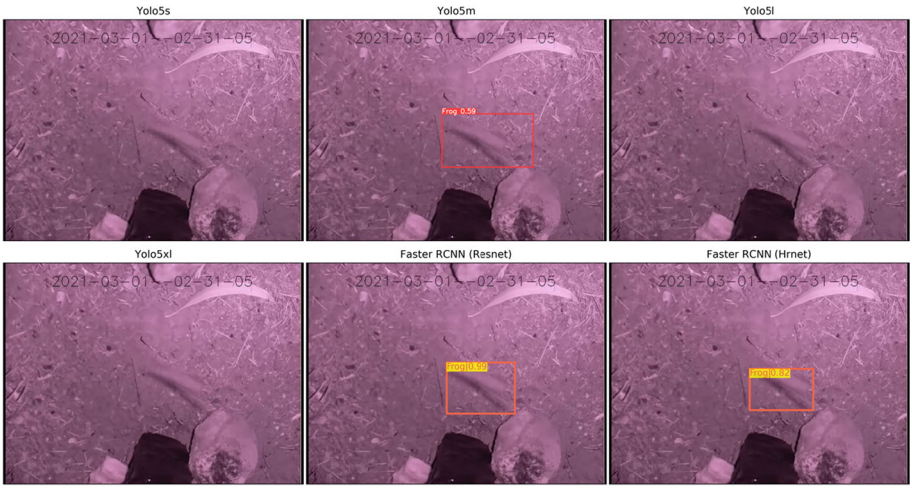

Fig. 15 Detection under fast motion. Only Faster RCNN and YOLOv5m are able to detect the frog in fast motion

5 Conclusions and future work

This paper presents a real-world dataset of small-sized animals, namely SAWIT. The dataset was collected from camera traps in realistic conditions for seven months between February 2021 and September 2021. Based on the collected data, we provided object-level annotations for 33,434 images and 34,820 animals into seven classes: frog, lizard, bird, small mammal, big mammal, spider, and scorpion. Compared with existing datasets, the SAWIT focuses on smaller fauna and includes ectotherms. To the best of our knowledge, this is the first annotated dataset with arthropods. The SAWIT covers practical challenges of wildlife data

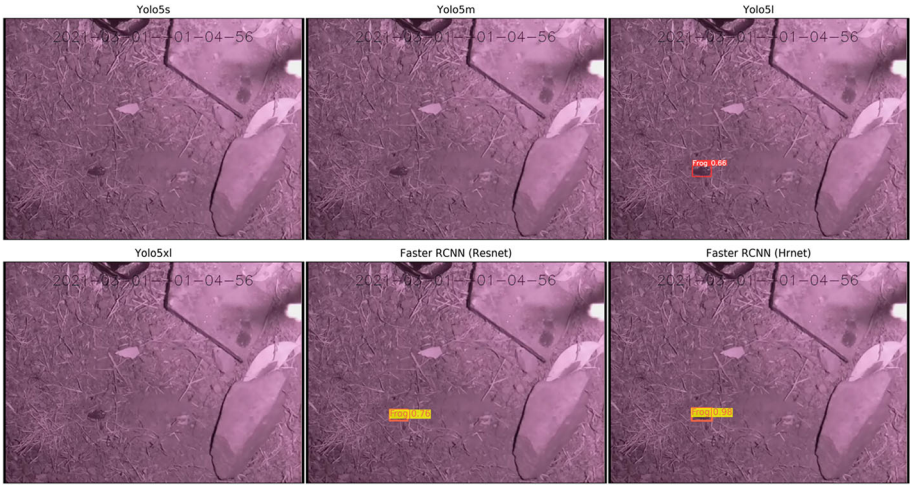

Fig. 16 Detection of tiny animals. This example shows an advantage of Faster RCNN with HRNet backbone in capturing details from imagery data

(e.g., cluttered backgrounds, low-quality images due to severe illuminations, occlusions, and fast motion) and can be considered as a benchmark for small-sized wildlife monitoring and conservation research. To demonstrate this capability, we experimented with state-of-the-art object detection algorithms (YOLO and Faster RCNN) and their variants for the task of animal detection on our dataset.

Experimental results show that spiders and birds can be well detected by both YOLO and Faster RCNN, while scorpions and small mammals remain difficult. We observed that missed detections of species such as lizards and scorpions are due to the indistinguishable appearance of the animals compared with their surroundings (e.g., a scorpion may camouflage into a background of grass and leaf litter, a lizard may look like a stick in dense vegetation). However, when the animals move, they can be easily noticed by human annotators. This suggests the use of temporal information in improving the object detection algorithms. We consider this direction for our future work. There is a trade-off between detection accuracy and computational speed, and the selection of a detection algorithm depends on the requirements of the downstream application. Based on our observations, we recommend the large model of YOLOv5, given its high detection accuracy (62.6% mAP) and real-time performance (83 fps).

Acknowledgements The project was made possible thanks to the contributions of Deakin University research networks program, DELWP and the Arthur Rylah Institute. Further testing and invaluable improvements were made by Brian Sala, scientist extraordinaire who also engaged as a citizen scientist in the project. The camera deployment and overall logistics were performed with the help of Lorenzo Galletta, Kendrika Gaur and Delaney Martin. Last but not least, the deployment, data collection and overall project would not have been possible without the contributions of our 30 citizen scientists affiliated with Land for Wildlife (https://www.wildlife.vic.gov.au/protecting-wildlife/land-for-wildlife). A huge thank you to all of them!

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data Availability Both the data and animal detection code are made available at https://github.com/dtnguyen0304/sawit.

Declarations

We have no conflict of interests to declare.

References

1. Beery S, Van Horn G, Perona P (2018) Recognition in terra incognita. In: European conference on computer vision, pp 472–489. https://doi.org/10.1007/978-3-030-01270-0_28
2. Beery S, Agarwal A, Cole E, et al (2021) The iWildCam 2021 competition dataset. arXiv:2105.03494, https://doi.org/10.48550/ARXIV.2105.03494
3. Clemann N (2015) Cold-blooded indifference: a case study of the worsening status of threatened reptiles from victoria, australia. Pac Conserv Biol 21(1):15–26. https://doi.org/10.1071/PC14901
4. Corcoran E, Denman S, Hanger J, et al (2019) Automated detection of koalas using low-level aerial surveillance and machine learning. Scientific Reports 9(3208). https://doi.org/10.1038/s41598-019-39917-5

5. Corva DM, Semianiw NI, Eichholtzer AC et al (2022) A smart camera trap for detection of endotherms and ectotherms. Sensors 22. https://doi.org/10.3390/s22114094
6. Deng J, Dong W, Socher R, et al (2009) ImageNet: A large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition, pp 248–255. https://doi.org/10.1109/CVPR.2009.5206848
7. Dundas SJ, Ruthrof KX, Hardy GESJ et al (2019) Pits or pictures: a comparative study of camera traps and pitfall trapping to survey small mammals and reptiles. Wildl Res 46(2):104–113. https://doi.org/10.1071/WR18074
8. Everingham M, Gool L, Williams CK et al (2010) The Pascal visual object classes (VOC) challenge. International J Comput Vis 88(2):303–338. https://doi.org/10.1007/s11263-009-0275-4
9. Fleming PJS, Meek PD, Ballard G et al (2014) Camera trapping: Wildlife management and research. CSIRO Publishing. https://doi.org/10.1071/9781486300402
10. Gagne C, Kini JR, Smith D, et al (2021) Florida wildlife camera trap dataset. In: IEEE Conference on computer vision and pattern recognition workshops, CV4Animals: Computer vision for animal behavior tracking and modeling workshop, pp 1–4
11. Gumbs R, Gray CL, Böhm M, et al (2020) Global priorities for conservation of reptilian phylogenetic diversity in the face of human impacts. Nature Communications 11(2616). https://doi.org/10.1038/s41467-020-16410-6
12. He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition. In: IEEE Conference on computer vision and pattern recognition, pp 770–778. https://doi.org/10.1109/CVPR.2016.90
13. Hodgson JC, Mott R, Baylis SM et al (2018) Drones count wildlife more accurately and precisely than humans. Methods Ecol Evol 9(5):1160–1167. https://doi.org/10.1111/2041-210X.12974
14. Horn GV, Aodha OM, Song Y, et al (2018) The INaturalist species classification and detection dataset. In: IEEE conference on computer vision and pattern recognition, pp 8769–8778. https://doi.org/10.1109/CVPR.2018.00914
15. Khan MH, McDonagh J, Khan S, et al (2020) AnimalWeb: A large-scale hierarchical dataset of annotated animal faces. In: IEEE conference on computer vision and pattern recognition, pp 6937–6946. https://doi.org/10.1109/CVPR42600.2020.00697
16. Kumar A, Kaur A, Kumar M (2019) Face detection techniques: A review. Artif Intell Rev 52(2):927–948. https://doi.org/10.1007/s10462-018-9650-2
17. Kuznetsova A, Rom H, Alldrin NG et al (2020) The open images dataset v4. Int J Comput Vis 128(7):1956–1981
18. Lawes MJ, Murphy BP, Fisher A et al (2015) Small mammals decline with increasing fire extent in northern australia: evidence from long-term monitoring in kakadu national park. Int J Wildl Fire 24(5):712–722. https://doi.org/10.1071/WF14163
19. Li S, Li J, Tang H, et al (2020) ATRW: A benchmark for Amur tiger re-identification in the wild. In: ACM international conference on multimedia, pp 2590–2598. https://doi.org/10.1145/3394171.3413569
20. Lin TY, Maire M, Belongie S, et al (2014) Microsoft COCO: Common objects in context. In: European conference on computer vision, pp 740–755
21. Liu C, Zhang R, Guo L (2019) Part-pose guided Amur tiger re-identification. In: IEEE International conference on computer vision workshop, pp 315–322. https://doi.org/10.1109/ICCVW.2019.00042
22. Martin SA, Rautsaw RM, Robb F et al (2017) Set AHDriFT: Applying game cameras to drift fences for surveying herpetofauna and small mammals. Wildl Soc Bull 41(4):804–809. https://doi.org/10.1002/WSB.805
23. Mathis A, Biasi T, Schneider S, et al (2021) Pretraining boosts out-of-domain robustness for pose estimation. In: IEEE Winter conference on applications of computer vision, pp 1858–1867. https://doi.org/10.1109/WACV48630.2021.00190
24. McShea W, Forrester T, Costello R et al (2016) Volunteer-run cameras as distributed sensors for macrosystem mammal research. Landsc Ecol 31(1):55–66. https://doi.org/10.1007/s10980-015-0262-9
25. Mohamed HED, Fadl A, Anas O et al (2020) MSR-YOLO: Method to enhance fish detection and tracking in fish farms. Procedia Comput Sci 170:539–546. https://doi.org/10.1016/j.procs.2020.03.123
26. Muksit AA, Hasan F, Hasan Bhuiyan Emon MF et al (2022) YOLO-Fish: A robust fish detection model to detect fish in realistic underwater environment. Ecol Inform 72(101):847. https://doi.org/10.1016/j.ecoinf.2022.101847
27. Ng XL, Ong KE, Zheng Q, et al (2022) Animal kingdom: A large and diverse dataset for animal behavior understanding. In: IEEE Conference on computer vision and pattern recognition, pp 19,001–19,012. https://doi.org/10.1109/CVPR52688.2022.01844
28. Nguyen DT, Li W, Ogunbona PO (2016) Human detection from images and videos. Pattern Recognit 51:148–175. https://doi.org/10.1016/j.patcog.2015.08.027

29. Nguyen H, Maclagan SJ, Nguyen TD, et al (2017) Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. In: IEEE International conference on data science and advanced analytics, pp 40–49. https://doi.org/10.1109/DSAA.2017.31
30. Norouzzadeh MS, Nguyen A, Kosmala M et al (2018) Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proc Natl Acad Sci 115(25):E5716–E5725. https://doi.org/10.1073/pnas.1719367115
31. OpenMMLab (2021) Mmdetection. https://github.com/open-mmlab/mmdetection
32. Parkhi OM, Vedaldi A, Zisserman A, et al (2012) Cats and dogs. In: IEEE Conference on computer vision and pattern recognition, pp 3498–3505. https://doi.org/10.1109/CVPR.2012.6248092
33. Rashid M, Broomé S, Ask K, et al (2022) Equine pain behavior classification via self-supervised disentangled pose representation. In: IEEE Winter conference on applications of computer vision, pp 152–162. https://doi.org/10.1109/WACV51458.2022.00023
34. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: IEEE Conference on computer vision and pattern recognition, pp 6517–6525. https://doi.org/10.1109/CVPR.2017.690
35. Ren S, He K, Girshick RB et al (2017) Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
36. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations, pp 1–14
37. Singh A, Pietrasik M, Natha G, et al (2020) Animal detection in man-made environments. In: IEEE Winter conference on applications of computer vision, pp 1427–1438
38. Stork NE (2018) How many species of insects and other terrestrial arthropods are there on earth? Annu Rev Entomol 63(1):31–45. https://doi.org/10.1146/annurev-ento-020117-043348
39. Sun K, Xiao B, Liu D, et al (2019) Deep high-resolution representation learning for human pose estimation. In: IEEE Conference on computer vision and pattern recognition, pp 5686–5696
40. Swanson A, Kosmala M, Lintott CJ et al (2015) Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. Sci Data 2:1–14. https://doi.org/10.1038/sdata.2015.26
41. Tabak MA, Norouzzadeh MS, Wolfson DW et al (2019) Machine learning to classify animal species in camera trap images: Applications in ecology. Methods Ecol Evol 10(4):585–590. https://doi.org/10.1111/2041-210X.13120
42. Tan M, Chao W, Cheng JK, et al (2022) Animal detection and classification from camera trap images using different mainstream object detection architectures. Animals 12(1976). https://doi.org/10.3390/ani12151976
43. Tingley R, Meiri S, Chapple DG (2016) Addressing knowledge gaps in reptile conservation. Biol Conserv 204:1–5. https://doi.org/10.1016/j.biocon.2016.07.021
44. Ultralytics (2021) Yolov5. https://github.com/ultralytics/yolov5
45. Vacavant A, Chateau T, Wilhelm A, et al (2012) A benchmark dataset for outdoor foreground/background extraction. In: Asian conference on computer vision workshops, pp 291–300. https://doi.org/10.1007/978-3-642-37410-4_25
46. Victoria Energy, Environment and Climate Action (2023) Bioregions and E V C benchmarks. https://www.environment.vic.gov.au/biodiversity/bioregions-and-evc-benchmarks
47. Weinstein BG, Garner L, Saccomanno VR et al (2022) A general deep learning model for bird detection in high-resolution airborne imagery. Ecol Appl 32(8):e2694. https://doi.org/10.1002/eap.2694
48. Winsen M, Denman S, Corcoran E, et al (2022) Automated detection of koalas with deep learning ensembles. Remote Sensing 14(10). https://doi.org/10.3390/rs14102432
49. Xu L, Jin S, Zeng W, et al (2022) Pose for everything: Towards category-agnostic pose estimation. In: European conference on computer vision, pp 398–416, https://doi.org/10.48550/arXiv.2207.10387
50. Yang S, Jeon S, Nam S, et al (2022) Dense interspecies face embedding. In: Conference on neural information processing systems, pp 1–14
51. Zhao ZQ, Zheng P, Xu ST et al (2019) Object detection with deep learning: A review. IEEE Trans Neural Netw Learn Syst 30(11):3212–3232. https://doi.org/10.1145/3484824.3484889

Authors and Affiliations

Thi Thu Thuy Nguyen[1] · Anne C. Eichholtzer[2] · Don A. Driscoll[2] · Nathan I. Semianiw[3] · Dean M. Corva[3] · Abbas Z. Kouzani[3] · Thanh Thi Nguyen[1] · Duc Thanh Nguyen[1]

1 School of Information Technology, Deakin University, Geelong, Australia

2 School of Life and Environmental Sciences, Deakin University, Melbourne, Australia

3 School of Engineering, Deakin University, Geelong, Australia